第 37 卷第 2 期 计算机应用研究 Vol. 37 No. 2
录用定稿 Application Research of Computers Accepted Paper



# 基于矩阵分解的 DeepWalk 链路预测算法 *

冶忠林 [1,3,4]，曹　蓉 [2,3,4]，赵海兴 [1,2,3,4†]，张　科 [2,3,4]，朱　宇 [2,3,4]

(1. 陕西师范大学 计算机学院，西安 710119; 2. 青海师范大学 计算机学院，西宁 810008; 3. 青海省藏文信息处理与机器翻译重点实验室，西宁 810008; 4. 藏文信息处理教育部重点实验室，西宁 810008)

**摘　要：**现有的链路预测方法的数据来源主要是基于邻居、路径、和随机游走的方法，使用的是节点相似性假设或者最大似然估计，尚缺少基于神经网络的链路预测研究。基于神经网络的一些研究表明，基于神经网络的 DeepWalk 网络表示学习算法可以更加有效地挖掘到网络中的结构特征，已有研究证明 DeepWalk 等同于分解目标矩阵。因此，提出了一种基于矩阵分解的 DeepWalk 链路预测算法（LPMF）。该算法首先基于矩阵分解的 DeepWalk 算法分解得到网络的表示向量；然后通过余弦相似度计算每对节点之间的相似度，构建目标网络的相似度矩阵；最后利用相似度矩阵，在三个真实的引文网络中进行链路预测实验。实验结果表明，提出的链路预测算法性能优于现存的 20 余种链路预测算法，这充分表明了 LPMF 能够有效地挖掘网络中节点之间的结构关联性，而且在实际网络的链路预测中能够发挥出较为优异的性能。

**关键词：**链路预测；神经网络；DeepWalk；网络表示学习；矩阵分解；相似度矩阵

**中图分类号：**TP301.6　　doi: 10.19734/j.issn.1001-3695.2018.07.0523

## Link prediction based on matrix factorization for Deepwalk

Ye Zhonglin[1,3,4], Cao Rong[2,3,4], Zhao Haixing[1,2,3,4], Zhang Ke[2,3,4], Zhu Yu[2,3,4]

(1. *College of Computer Science*，*Shaanxi Normal University*, *Xi'an* 710119, *China*; 2. *College of Computer Science*，*Qinghai Normal University*, *Xining* 810008, *China*; 3. *College of Computer*，*Qinghai Normal University*, *Xining* 810008, *China*; 4. *Tibetan Information Processing & Machine Translation Key Laboratory of Qinghai Province*, *Xining* 810008, *China*)

**Abstract:** The data sources of existing link prediction algorithms are mainly based on neighbors, paths, and random walk methods, the link prediction algorithms use mainly node similarity assumptions or maximum likelihood estimates. The link prediction based on neural network is still absent. Some research achievements based on neural network show that the DeepWalk algorithm based on neural network is an efficient network representation learning algorithm, which can more effectively learn the network structure features in the network. It has been proven that DeepWalk is equivalent to factorize the target matrix. Therefore, this paper presents a link prediction algorithm (LPMF) based on matrix factorization of DeepWalk. This algorithm based on matrix factorization uses the DeepWalk algorithm to get the network representation vectors. And then, the similarities between node pairs of nodes are calculated by the cosine similarity method. Based on that, the similarity matrix of the target network is constructed. Finally, we use the similarity matrix to conduct the link prediction experiments on three real-world citation networks. The experimental results show that the link prediction algorithm proposed in this paper is superior to the existing 20 kinds of link prediction algorithms, which fully shows that LPMF can effectively find the structural correlation between nodes in the network, and performs a more excellent performance in the actual tasks of link prediction.

**Key words:** link prediction; neural network; deepwalk; network representation vectors; matrix factorization; similarity matrix

## 0　引言

随着网络技术的不断发展，复杂网络的演化已经成为当前复杂网络研究领域中的热点问题,而链路预测又是网络演化及建模中一个基本的计算问题。网络中的链路预测是指基于已知的网络结构等信息预测网络中尚未产生连边的两个节点之间产生连接的可能性[1]。预测已经存在但尚未被发现的连接关系被称为对未知的预测，而预测节点之间未来可能产生的连边被称为对未来的预测。近年来，大规模网络中的链路预测已经成为一个研究热点，而链路预测的成果也被应用到各类任务中，如网络建模[2~5]、蛋白质网络预测[6,7]、社交网络分析[8~10]、标签分类[11~13]、知识获取[14]、异常检测[15~17]、推荐系统[18,19]等。为了揭示真实世界的网络演化的机制，也提出了各类型的网络建模方法[20,21]，但是非常难以判断何种网络建模方法能够反映真实网络的生成过程。受益于计算性能的提升和大规模社交网络数据的公开访问，链路预测的发展经历了一个从节点属性挖掘到网络属性挖掘的过程，以及从小规模网络链路预测到大规模社交网络链路预测的发展经

**收稿日期：**2018-07-14；**修回日期：**2018-09-28　**基金项目：**NSFC 支持项目（11661069，61763041）；国家教育部长江学者和创新团队发展计划资助项目（IRT_15R40）；青海省自然科学基金资助项目（2013-Z-Y17，2014-ZJ-721）；中央高校基本科研基金资助项目（2017-TS-045）

**作者简介：**冶忠林（1989-），男，博士研究生，主要研究方向为自然语言处理、知识表示学习；曹蓉（1994-），女，陕西眉县人，硕士研究生，主要研究方向为复杂网络、链路预测；赵海兴（1969-），男（通信作者），教授，博导，主要研究方向为复杂网络、超图理论（845172605@qq.com）．

历[22]。但是，传统的基于熵或者最大似然估计[23]的链路预测算法具有很大的计算复杂度和不精准度[24]。而且，目前尚缺乏适合于大规模数据集的高效链路预测算法以及对于大规模真实数据在应用层面的深入分析和研究。这两方面的研究有助于揭示链路预测这个问题本身存在的优势与局限性。

Google 提出的 word2vec[25,26]是基于三层神经网络概率模型的一种词语表示学习算法。该算法基于大规模语言语料，使用神经网络算法获得每个词语在语言空间中低纬地、稠密地向量表示形式。使用固定的窗口大小，获取当前词语在窗口内相邻的词语作为它的上下文词语，然后将当前词和它的上下文词输入到神经网络中学习。基于神经网络的词语表示学习在语言模型中取得了巨大的成功。随后在网络空间模型中，基于 word2vec 算法，DeepWalk[27]网络表示学习算法被提出。该算法使用了随机游走的过程获取当前节点的上下文节点，然后将当前节点和它的上下文节点输入到神经网络模型中进行学习，最终获得每个节点在网络空间模型中的低纬的、稠密的向量表示形式。网络表示学习算法其实质是将网络特征转换为便于处理的向量形式。将获得的向量可用可视化的方法展示在 2 维的平面上，展现具有相似属性的节点所具有的聚类现象，如图 1 所示。

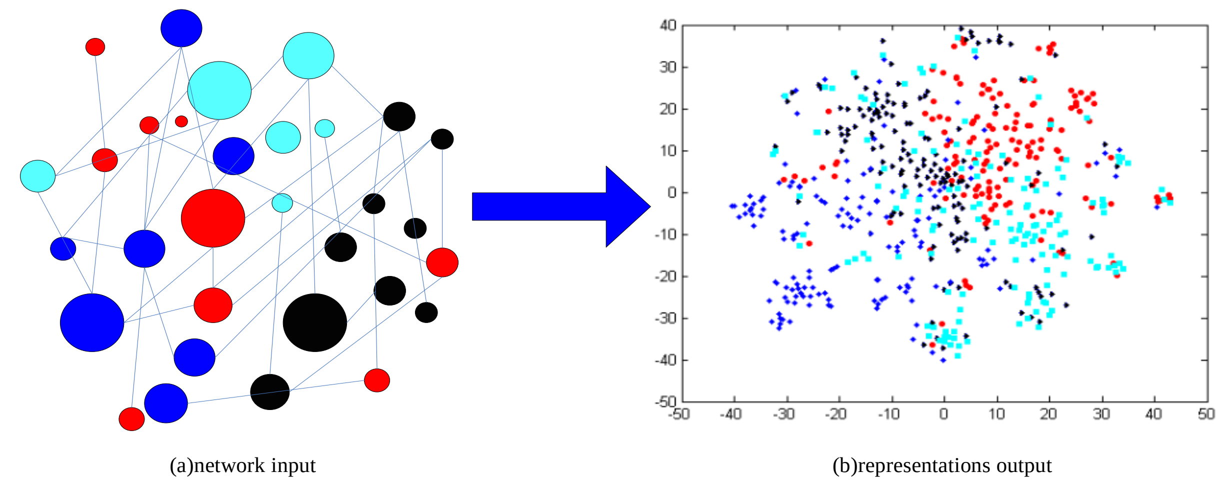



图 1 网络表示学习可视化示例

Fig. 1 Visualization case of network representation learning

DeepWalk 网络表示学习是基于神经网络的算法，通过对网络的结构的深入学习，使得具有相似网络结构的节点具有相似的网络表示向量。使用 DeepWalk 网络表示学习方法不仅可以帮助人们更好地理解网络中节点间的结构关联性，还可以进一步缓解由于网络稀疏性所造成的训练数据不足问题。因为 DeepWalk 采用的是局部随机游走，在大规模网络结构挖掘中，DeepWalk 算法具有更高效的特点。随后清华大学的 Yang 等人[28]从数学角度证明了证 DeepWalk 方法等同于分解一个目标矩阵 $\boldsymbol{M}$，但没有进行实验验证两种表示学习方法之间的差异。由于采用不同的矩阵分解算法可得到不同的网络表示学习结果。因此，DeepWalk 算法和分解矩阵 $\boldsymbol{M}$ 都能获得网络的表示特征。这两者之间的区别是 DeepWalk 使用随机游走策略避免直接计算和分解矩阵 $\boldsymbol{M}$，能够适用于大规模的网络表征学习。分解一个目标矩阵 $\boldsymbol{M}$ 具有较高的时间复杂度，且算法精度受限于分解算法的效率。

本文基于 DeepWalk 等同于矩阵分解的研究工作，提出了一种基于矩阵分解的链路预测算法。该方法首次将基于矩阵分解的 DeepWalk 表示学习方法引入到网络的链路预测过程中，验证了类神经网络的方法可更有效地挖掘网络中的结构关联性，训练得到的网络表示在链路预测实证中也能发挥较为出色的表现。不同于传统的在链路预测中使用的全局随机游走[29]、有重启的随机游走[30]、局部随机游走[31]算法，DeepWalk 算法不仅使用了局部随机游走获得了节点的上下文节点，而且把当前节点和上下文节点一起输入到神经网络中进行训练，彻底并深入地挖掘出了网络的结构特征，反映出了节点之间的结构相似性。因为 Yang 等人[28]证明了 DeepWalk 等同于矩阵分解矩阵 $\boldsymbol{M}$，因此，本文中使用的基于矩阵分解的 DeepWalk 表示学习算法可避免网络中随机游走和进行神经网络学习和训练的过程，而是采用高效地矩阵分解方法对目标矩阵 $\boldsymbol{M}$ 进行分解即可。该方法能延续 DeepWalk 的优点，同时也满足了从邻接矩阵直接转换为网络表示形式的需求。

综上，本文的主要贡献有如下两点：a）将基于矩阵分解的 DeepWalk 网络表示学习引入到网络的链路预测，即使用简单的矩阵分解也可以达到与神经网络算法几乎等同的预测能力；b）本文基于三个真实的引文网络数据集进行了链路预测、可视化、案例研究实验。实验结果表明，本文引入的方法可有效地学习到网络的结构特征，使得网络具有更好的预测功能。

## 1 相关工作

对于链路预测问题，目前，常用的方法是基于节点相似性的链路预测算法。该类方法主要有局部信息的相似性指标、基于路径的相似性指标和基于随机游走的相似性指标三种指标。

**基于局部信息的相似性指标**包括基于共同邻居的相似性指标（CN）[32]、AA 指标(Adamic-Adar)[33]和资源分配指标 RA（resource allocation）[34]。CN 指标是最简单的基于节点局部信息的相似性。其定义为：若两个节点拥有很多的共同邻居，那么这两个节点相似。共同邻居数越多，则它们的相似性就越高。在共同邻居的基础上，从不同角度考虑节点度对其影响，可细分为 6 种相似性指标，分别为余弦相似性指标（Salton 指标）[35]、Jaccard 指标[36]、Sorenson 指标[37]、大度节点有利指标（HPI）[38]、大度节点不利指标（HDI）[18]以及 LHN-I 指标[39]。基于共同邻居的相似性指标的优势在于其计算复杂度较低，适合于大规模的网络应用，但是由于使用的信息有限，导致其算法预测出的精确度受到限制。

**基于路径的相似性指标**分别是局部路径相似性指标

(LP)、Katz 指标[40]和 LHI-II 指标[39]。LHI-II 指标中若两个节点所连接的节点之间相似，则这两个节点也相似，即使它们之间没有共同的邻居节点。该方法在建立训练集的时候往往需要大量可靠的标签属性，因此，对于未标注的网络其扩展能力较差。基于路径的相似度指标随着网络规模的增大和考虑的路径的长度 $n$ 的增长，路径指标的计算复杂度也越来越大。LP 算法的路径长度趋向于无穷大时，可认为 LP 算法相当于考虑了网络全部路径的 Kata 算法。此时 LP 算法的计算量可采用矩阵求逆的方式得到有效的降低。Kata 算法考虑的全部的路径信息，但是由于高阶的路径对相似度的贡献很小，因此也是采用矩阵求逆的方法获得相似度。关于矩阵求逆一般是采用稀疏矩阵求逆的快速算法。

**基于随机游走的相似性指标**包括平均通勤时间(ACT)[29]、余弦相似性指标(Cos+)[41]、局部随机游走(LRW)[42]及有叠加效应的随机游走指标(SRW)[42]。ACT 算法认为两个节点的平均通勤时间越小，那么两个节点越靠近。通勤时间就是一个随机粒子从一个节点到达另外一个节点再返回到起始节点的平均步数。Cos+采用了玛氏距离来衡量两个节点的向量之间的不相似度。因为 ACT 是一种全局随机游走，而且全局随机游走往往有很高的计算复杂度，很难应用于大规模的网络中，因此刘伟平等[42]等提出了局部随机游走，LRW 考虑了有限步数内的随机游走，因此算法的计算复杂度要低很多。在 LRW 的基础上，将前面的结果与最后一步的结果相加就得到了 SRW。SRW 给邻接的节点给予了更多的机会与目标节点相连接。因此，SRW 是一种充分考虑了真实网络特征的算法。

当然，还有一些其他类型的基于节点相似性的链路预测算法，比如基于图理论的矩阵森林指数（MFI）[43]、自洽相似性指数（TSCN）[44]、基于偏好的相似性指标（PA）[45]、基于朴素贝叶斯模型的指标（LNBAA、LNBCN、LNBRA）[46]等。传统的基于共同邻居的指标算法不会考虑共同邻居的权重信息。然而不同的邻居对整个网络的影响力是不一样的，因此，刘震等人引入了一个角色权重函数，用于计算不同邻居的影响力大小[46]。并将角色函数引入到 AA、CN、RA 中，提出了基于朴素贝叶斯模型的算法。

自从 Moore 和 Newman 在 2008 年发表的《自然》论文[47]，以及 Redner 在《自然》上的评论文章[48]。链路预测就一直是复杂网络研究的重点，也取得了很多的成功。以上的链路预测算法均是采用统计方法获得节点之间的相似度值，其中的一些算法均表现出了优异的性能。而目前，一些基于神经网络的算法可以更加高效的获得网络的特征向量，基于该特征向量也可以进行各类机器学习任务，比如链路预测等。DeepWalk 首先将该思路引入到了网络链路预测任务中，并在公开的真实数据集上表现出了优异的预测性能。本文通过引入 DeepWalk 的实质为分解网络特征矩阵的依据，首次验证了基于矩阵分解的 DeepWalk 链路预测算法的可行性。当然，网络表示学习有很多，例如，TADW[49]、MMDW[50]和 NEU[51]等，这些网络表示算法的性能均比 DeepWalk 优异，但本文的研究目标在于研究采用矩阵分解方法达到和神经网络同样的链路预测性能，并非对基于神经网络的网络表示算法进行横向对比。

## 2 本文方法

本文基于矩阵分解的 DeepWalk 方法，获得网络中的每个节点向量表示，然后使用余弦相似度计算方法构建出了网络节点的相似度矩阵。最后，将相似度矩阵应用到了网络的链路预测中，并通过计算其 AUC 指标，进而验证所提出 LPMF 算法的可行性和有效性。在详细的解释本文的方法之前，首先详细介绍了基于矩阵分解的 DeepWalk 方法，详细证明了 DeepWalk 算法就是矩阵分解一个目标矩阵 $\boldsymbol{M}$，该方法也是本文所提出的方法的基础。

### 2.1 基于矩阵分解 DeepWalk 算法

SGNS 主要应用于语义网络中，词语之间只有上下文关系，上下文用窗口确定。SGNS 就是将（词语，上下文）对进行收集，之后输入到一个三层的浅层神经网络中进行训练。初始状态时，给每个词语定义一个任意的向量表示，然后随着（词语，上下文）在神经网络中的重现，不断的调整词语的向量表示。SGNS 算法能够充分的利用上下文信息训练词语之间的语义关联。

受到 SGNS 算法的启发，DeepWalk 算法对 SGNS 算法做了部分的修改，使得算法从语义网络迁移到各种普通的网络中，比如社交网络等。这种泛化使得表示学习算法能够被应用在各种网络中，从而得到更普遍的应用。在 DeepWalk 中，改变的仅仅是上下文的获取方式，SGNS 采用滑动的窗口获取上下文，而 DeepWalk 采用的是随机游走的方式获取上下文，其他的都未改变。同样使用（当前节点，上下文节点）对输入到一个三层的浅层神经网络中。即 DeepWalk 和 SGNS 改变的仅仅是上层的输入，底层的算法都未曾改变。

随后，Levy 等人[52]证明了 SGNS 词向量表示学习算法就是矩阵分解一个 SPPMI 矩阵，简称为 $\boldsymbol{M}$ 矩阵，矩阵 $\boldsymbol{M}$ 的表达式为

$$M_{i,j}=\log\frac{N(v_i,c_j)\cdot|D|}{N(v_i)\cdot N(c_j)}-\log n\text{。}\tag{1}$$

其中: $n$ 为每个（词语，上下文）对的负采样个数。$D$ 表示整个训练集中函数的词语数量。$N(v)$ 表示词语 $v$ 在整个训练集 $D$ 中出现的次数，$N(c)$ 表示上下文词语 $c$ 在整个训练集 $D$ 中出现的次数，$N(v,c)$ 表示（词语，上下文）对在整个训练集中出现的次数。

受到 SGNS 算法的启发，Yang 等人[28]从数学角度证明了 DeepWalk 算法类似于分解 SGNS 的特征矩阵，即矩阵分解一个目标矩阵 $\boldsymbol{M}$，矩阵 $\boldsymbol{M}$ 的表达式为

$$M_{ij}=\log\frac{N(v_i,c_j)}{N(v_i)},\tag{2}$$

此时的 $v_i$ 表示为网络中的节点，而非词语，$c_j$ 表示为当前节点的上下文节点。上下文节点通过随机游走获取。对于网络 $G=(V,E)$，$V$ 为网络 $G$ 的顶点集，$E$ 为网络 $G$ 的边集。此时设 $D$ 为随机游走序列中生成的（当前节点，上下文节点）的集合，其中 $D$ 的每个实体均为一个上下文节点对 $(v,c)$。

假设随机游走的步长为 $t$，那么在集合 $D$ 中，节点 $i$ 被访问的次数为 $2t$ 次。因为 $N(v_i)/|D|$ 表示节点 $v_i$ 在随机游走过程中出现的次数，该值恰好和节点 $v_i$ 的 PageRank 的值相等。另外，$2tN(v_i,c_j)/N(v_i)$ 表示在节点 $v_j$ 在节点 $v_i$ 周围且在随机游走步长为 $t$ 以内出现的次数。本文定义 PageRank 的转移矩阵为 $\boldsymbol{A}$，并将节点 $i$ 的度记为 $d_i$，于是就有

$$A_{ij}=\begin{cases}1/d_i & (i,j)\in E\\ 0 & otherwise\end{cases}。\quad(3)$$

令 $e_i$ 表示一个 $|V|$ 维的行向量，只有第 $i$ 列元素为 1，其余全为 0。假设从节点 $i$ 开始游走，并用 $e_i$ 来表示初始状态，则 $e_iA$ 为节点 $i$ 相对于所有节点的一个空间分布，$e_iA$ 中的第 $j$ 个实体值表示游走粒子从节点 $i$ 游走到节点 $j$ 的概率大小。依此类推，$e_iA^t$ 中的第 $j$ 个实体值表示游走粒子在 $t$ 之内节点 $i$ 游走到节点 $j$ 的概率大小。以此可以得到 $\left[e_i(A+A^2+A^3+...+A^t)\right]_j$ 表示节点 $v_j$ 在节点 $v_i$ 周围且在随机游走步长为 $t$ 以内出现的次数。综上可以计算出：

$$\frac{N(v_i,v_j)}{N(v_i)}=\frac{\left[e_i(\boldsymbol{A}+\boldsymbol{A}^2+\boldsymbol{A}^3+...+\boldsymbol{A}^t)\right]_j}{t},\quad(4)$$

因此，

$$M_{ij}=\log(\left[e_i(\boldsymbol{A}+\boldsymbol{A}^2+\boldsymbol{A}^3+...+\boldsymbol{A}^t)\right]_j/t)。\quad(5)$$

计算出 $\boldsymbol{M}$ 的时间复杂度为 $O(|V|^3)$，实际上，DeepWalk 算法采用随机游走的采样方法来避免准确地计算矩阵 $\boldsymbol{M}$。而矩阵分解的方法不可避免的要计算出 $\boldsymbol{M}$ 以便于进行分解。因此，Yang 等人[28]权衡了算法速度与精确度两方面，得到分解到的目标矩阵为：$\boldsymbol{M}=(\boldsymbol{A}+\boldsymbol{A}^2)/2$，当网络是稠密网络时，甚至直接可以分解矩阵 $\boldsymbol{A}$，即 $\boldsymbol{M}=\boldsymbol{A}$。因为，相比于 $\boldsymbol{M}$，在 $\log\boldsymbol{M}$ 矩阵中含有更多的非零元素，而 Yu 等人[53]已经证明了，在矩阵分解时，使用平方损失评估函数，分解的时间复杂度与矩阵中含有的非零元素成正比例关系。在本文中，通过分解矩阵 $\boldsymbol{M}$ 而不是分解 $\log\boldsymbol{M}$ 来提高算法的效率。因此，本文的算法复杂度主要来自两部分，一部分是来自构建 $\boldsymbol{M}=(\boldsymbol{A}+\boldsymbol{A}^2)/2$，另外一分部是分解矩阵 $\boldsymbol{M}$。构建邻接矩阵 $\boldsymbol{A}$ 的算法时间复杂度为 $O(n^2)$，如果使用 SVD 分解该目标矩阵 $\boldsymbol{M}$，则分解部分的时间复杂度为 $O(n^3)$。

## 2.2 基于矩阵分解的链路预测

基于矩阵分解的 DeepWalk 链路预测算法是建立在矩阵分解的基础之上，使用不同的分解方法获得不同的网络表示。给定一个网络的连边表示形式，可以将该网络转换为网络的邻接矩阵形式，基于该邻接矩阵，可以生成网络的矩阵分解的目标矩阵 $\boldsymbol{M}$。本文中，使用矩阵方法将目标矩阵 $\boldsymbol{M}$ 分解为三个矩阵的相乘的形式。关于矩阵分解算法，本文中使用的是 SVDS 算法，该算法相对于 SVD 算法有以下的优势：a）SVDS 是 SVD 算法的变体，都是基于奇异值分解的方式分解矩阵，但是计算复杂度降低了;b）SVDS 可以返回指定个数的最大的特征值以及其特征行向量和列向量;c）相比于 SVD 算法，SVDS 算法则具有更强的可定制性和塑造性。基于以上的优点，本文中使用 svds 算法分解目标矩阵 $\boldsymbol{M}$。

本文中，基于真实网络数据集的链路预测算法的具体框架如图 2 所示。

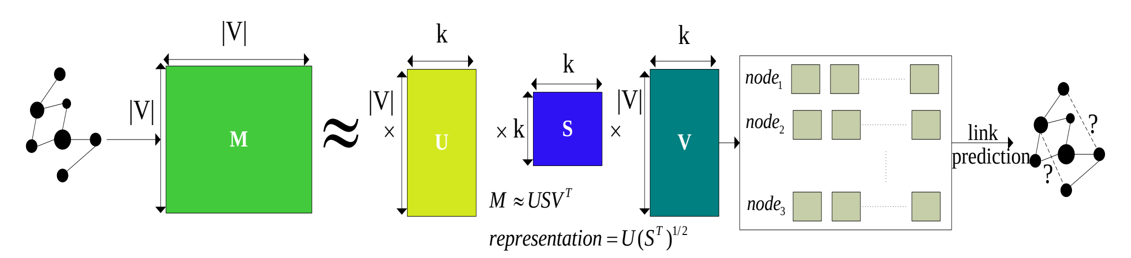



图 2 LPMF 算法框架图

Fig. 2 Algorithm framework of LPMF

如图 2 所示，本文中的框架可具体分解为五个步骤，每个步骤的任务处理如下所示：

a)输入一个由边集组成的网络，然后将该网络分割为训练集合测试集，将训练集转换为邻接矩阵 $\boldsymbol{A}$，然后基于该邻接矩阵 $\boldsymbol{A}$，求得该网络所需要分解的目标矩阵 $M=(A+A^2)/2$。

b)使用 SVDS 分解算法，将目标矩阵 $\boldsymbol{M}_{|V|\times|V|}$ 分解为 $\boldsymbol{U}_{|V|\times k}$，$\boldsymbol{S}_{k\times k}$ 和 $\boldsymbol{V}_{k\times|V|}$ 三个矩阵的相乘形式。

c)根据将目标矩阵 $\boldsymbol{M}_{|V|\times|V|}$ 分解的到三个矩阵 $\boldsymbol{U}_{|V|\times k}$，$\boldsymbol{S}_{k\times k}$ 和 $\boldsymbol{V}_{k\times|V|}$，可将矩阵 $\boldsymbol{U}_{|V|\times k}$ 和 $\boldsymbol{S}_{k\times k}$ 相乘，得到网络中每个节点的网络表示形式，即，网络中每个节点的表示组成的矩阵为 $\boldsymbol{U}_{|V|\times k}\times(\boldsymbol{S}^T{}_{k\times k})^{1/2}$，该矩阵是一个 $|\boldsymbol{V}|$ 行 $k$ 列的矩阵表示。

d)基于计算所得的节点的表示，使用余弦相似度计算方法，计算每两个节点之间的余弦相似度，之后再构建一个 $|\boldsymbol{V}|$ 行 $|\boldsymbol{V}|$ 列的节点相似度矩阵。

e)基于网络的节点相似度矩阵和测试集，使用 AUC 指标，评估本文所提出方法的链路预测性能。

SVDS 奇异值分解的目标是用三个子矩阵相乘来表示一个复杂的矩阵。对于任意一个矩阵都可以使用 SVDS 分解方法，即在本文中，对于一个 $m\times n$ 的矩阵 $M$，存在如下的 SVDS 分解：

$$M_{|V|\times|V|}\approx U_{|V|\times k}S_{k\times k}(V_{k\times|V|})^T$$

其中:$|V|$ 表示网络中节点的个数。$k$ 为特征值的个数，在本文中可被认为是向量的长度大小。矩阵 $U$ 是 $M$ 的奇异向量，$S$ 是一个对角矩阵，其中的元素为 $M$ 奇异值，$MM^T$ 的正交单位特征向量组成 $U$，特征值组成 $\boldsymbol{S}^T\boldsymbol{S}$，$MM^T$ 的正交单位特征向量组成 $\boldsymbol{V}$，特征值组成 $\boldsymbol{SS}^T$。svds 被广泛应用于各类数据降维、推荐系统等任务中。

以上 5 个步骤完整的构成了本文算法的主要流程。为了更加详细的展示细节，本文中提供了如下的算法伪代码。

**Algorithm:** LPMF (G, train-ratio, $k$)

Input：

Network edge set：$G$

Train ratio of dataset：training-ratio

Representation length: $k$

Output: AUC

① Get the edge set of the network $G$

② Count the amount of nodes，named as $|V|$

③ Split the network G into training set and testing set：

$$[training\ set,\ testing\ set] \xleftarrow{training\ ratio} G$$

④ Initial the adjacency matrix $\boldsymbol{A}$ for training set

⑤ Initial the target matrix $\boldsymbol{M}$:

$\boldsymbol{M} = (\boldsymbol{A}+\boldsymbol{A}^2)/2$

⑥ Factorize the matrix $\boldsymbol{M}$:

$[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = \text{svds}(\boldsymbol{M}, k)$

Representations matrix $\boldsymbol{R} = \boldsymbol{U} \times (\boldsymbol{S}^T)^{1/2}$

⑦ Compute cosine similarity for each node pairs

$s = \text{sim}(\boldsymbol{a},\boldsymbol{b}) = (\boldsymbol{a}*\boldsymbol{b}) / (||\boldsymbol{a}|| * ||\boldsymbol{b}||)$

⑧ Build the similarity matrix $\boldsymbol{S}$:

$$s_{i,j} \in \boldsymbol{S},\ \ \forall i, j \le |V|$$

⑨ Compute AUC using testing set:

$$AUC \longleftarrow training\ set$$

## 3 实验结果与分析

### 3.1 实验设置

本文中所采用的实验数据集均为真实的引文网络数据集，关于数据集的详细情况如表 1 所示。本文所使用的三个数据集为 Citeseer，DBLP，Cora。三个数据拥有几乎相同的网络节点，都为 3000 左右的节点个数。但是边的个数不一致，Citesser 和 Cora 数据集拥有几乎相同的边数量，但是 DBLP 数据集中边的个数几乎是其他数据集的 6 倍大小。另外，还可以发现，在几乎拥有相同节点数量的情况下，边的数量越多，网络的密度越大，同时网络的平均度大小也越大。如果边数和节点数几乎相同，则网络的密度也几乎一样。虽然，DBLP 和 Cora 数据集的边数差别很大，但是他们拥有几乎一样的网络直接和平均路径长度。根据网络的平均度和密度，Citeseer 和 Cora 网络是一个稀疏网络，而 DBLP 网络是一个稠密网络。

表 1 数据集描述

Table 1 Dataset descriptions

| 数据集 | 节点数 | 边数 | 类别数 | 平均度 | 网络直径 | 平局路径长度 | 密度 | 平局聚类系数 |
|---|---|---|---|---|---|---|---|---|
| Citeseer | 3312 | 4732 | 6 | 2.857 | 8 | 2.02 | 0.001 | 0.080 |
| DBLP | 3119 | 39516 | 4 | 21.07 | 17 | 4.71 | 0.005 | 0.221 |
| Cora | 2708 | 5429 | 7 | 4.01 | 15 | 4.79 | 0.001 | 0.130 |

### 3.2 实验结果分析

本文首先采用 SVDS 矩阵分解方法将 $M=(A+A^2)/2$ 矩阵分解为 $\boldsymbol{U,S,V}$ 三个矩阵。用 $\boldsymbol{U}\times(\boldsymbol{S}^T)^{1/2}$ 来表示网络中每个节点的向量表示。然后基于余弦相似度计算方法构建网络节点的相似度矩阵，并在 Citesser、DBLP 和 Cora 三个数据集上做了实验仿真。为了验证本文所提出的方法的有效性，使用了相关工作章节中所列出的所有方法进行了对比。在相关工作章节列举的链路预测算法多为采用统计的方法获得节点之间的相似度值，本文提出的基于矩阵分解的 DeepWalk 链路预测算法采用了类神经网络方法获得了网络的结构特征矩阵，之后采用矩阵分解算法构建了节点之间的相似度值。故本文提出的 LPMF 方法仅在特征获取方面参考了 DeepWalk 方法，但是实质还是一个矩阵分解算法。因此，本文提出的 LPMF 方法和相关工作中列举的方法之间具有可比性。在本实验中，本文设置了训练得到的表示向量的长度大小为 100，并设置训练集的训练比例为 0.7，0.8 和 0.9。实验结果如下表 2 所示：

从表 2 可以发现，利用 LPMF 算法与 21 种比较常用的链路预测算法进行了对比，通过数据分析发现，在 Citeseer、DBLP 和 Cora 数据集上，虽然 LPMF 从结构上挖掘出了有效的网络特征，但是实验结果表明，基于 MFI 的网络特征挖掘更能体现出网络的增长本质。本文提出的 LPMF 算法和 Katz 算法性能几乎相同。Katz 是基于全局路径统计的算法，在平均路径长度较短的网络中性能突出，LPMF 算法能够在较长的平均路径上随机游走获得更多的网络特征。因此，在 Citeseer 数据集上，Kata 算法优于 LPMF 算法，在 DBLP 和 Cora 数据集上，LPMF 算法性能优于 Katz 算法。总之，本文提出的 LPMF 算法性能优于其余 19 种链路预测算法，因为本文充分的利用了网络结构特征构建网络表示向量。

表 2 citesser, dblp 和 cora 数据集上链路预测结果

Table 2 The results of link prediction on Citeseer, DBLP and Cora

| 数据集 | Citeseer | | | DBLP | | | Cora | | |
|---|---|---|---|---|---|---|---|---|---|
| 训练率 | 0.7 | 0.8 | 0.9 | 0.7 | 0.8 | 0.9 | 0.7 | 0.8 | 0.9 |
| CN | 68.13 | 72.08 | 74.67 | 85.49 | 88.40 | 90.68 | 69.50 | 72.38 | 78.19 |
| Salton | 66.32 | 72.73 | 74.44 | 86.00 | 87.92 | 90.74 | 69.38 | 72.13 | 77.89 |
| Jaccard | 66.51 | 72.25 | 74.33 | 85.92 | 88.26 | 90.98 | 69.25 | 72.00 | 77.09 |
| HPI | 66.29 | 72.18 | 74.42 | 85.61 | 88.95 | 90.77 | 69.38 | 72.44 | 77.93 |
| HDI | 66.03 | 72.52 | 74.17 | 85.72 | 88.31 | 90.84 | 69.52 | 72.53 | 76.67 |
| LHN-I | 66.47 | 72.93 | 74.46 | 85.80 | 87.87 | 89.95 | 69.19 | 72.16 | 77.30 |
| AA | 66.37 | 72.22 | 74.33 | 86.00 | 88.22 | 90.95 | 69.35 | 72.66 | 77.60 |
| RA | 66.37 | 72.12 | 74.63 | 86.56 | 88.50 | 90.81 | 69.47 | 72.47 | 77.97 |
| PA | 78.98 | 79.06 | 79.53 | 76.39 | 77.13 | 77.54 | 71.50 | 71.91 | 71.50 |
| LP | 81.06 | 86.83 | 88.45 | 92.96 | 93.65 | 94.94 | 80.12 | 82.97 | 87.90 |
| Katz | 96.89 | 97.98 | 97.19 | 93.45 | 94.18 | 94.83 | 90.89 | 92.14 | 94.44 |
| LHNII | 95.76 | 96.85 | 96.20 | 90.86 | 91.80 | 92.80 | 89.41 | 90.37 | 93.64 |
| LNBAA | 66.37 | 72.64 | 74.52 | 86.07 | 88.42 | 91.12 | 69.42 | 72.50 | 78.01 |
| LNBCN | 66.70 | 72.27 | 74.25 | 85.60 | 88.47 | 90.80 | 69.50 | 72.19 | 77.79 |
| LNBRA | 66.05 | 72.23 | 74.27 | 85.86 | 88.91 | 91.23 | 69.32 | 72.84 | 77.74 |
| ACT | 75.88 | 75.59 | 73.79 | 79.00 | 80.07 | 80.84 | 74.11 | 73.67 | 74.00 |
| Cos+ | 88.57 | 89.38 | 88.49 | 91.53 | 93.47 | 95.08 | 90.25 | 90.98 | 93.22 |
| LRW | 87.21 | 90.13 | 91.25 | 92.75 | 93.35 | 94.09 | 88.48 | 90.58 | 93.63 |
| SRW | 86.34 | 90.05 | 90.47 | 90.50 | 92.25 | 94.06 | 88.40 | 90.50 | 93.62 |
| MFI | 96.68 | 98.00 | 97.80 | 95.13 | 96.00 | 97.07 | 93.13 | 94.25 | 95.60 |
| TSCN | 84.26 | 85.68 | 86.27 | 91.25 | 91.03 | 92.34 | 88.35 | 90.64 | 92.98 |
| LPMF | 87.18 | 90.64 | 94.98 | 93.42 | 94.70 | 95.13 | 89.57 | 92.13 | 93.93 |

### 3.3 分布可视化

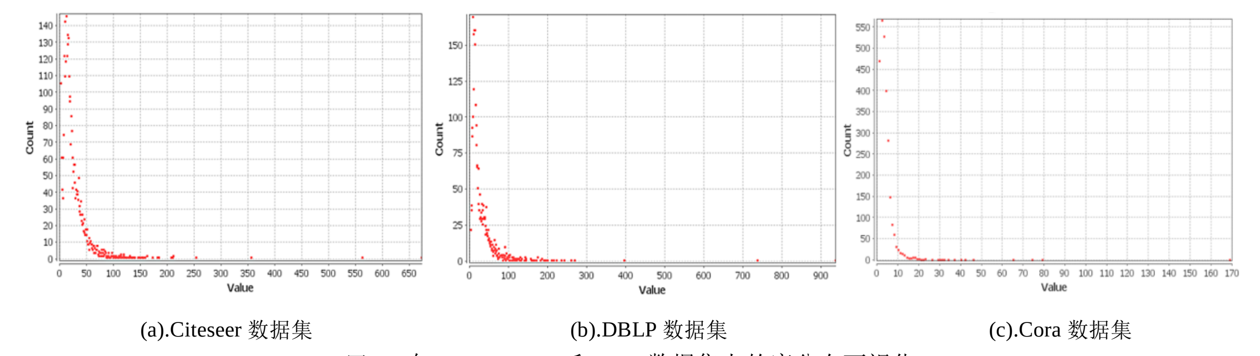
网络中最基本也是最重要的参数就是顶点的度。网络的最基本的性质之一就是网络的度分布，即网络中顶点度的频率分布。网络的度分布与网络的拓扑结构密切相关。因此，可以根据网络的度分布来基本确定网络的类型。比如，大多数网络具有无标度性，其幂律分布完全是由度分布指数来确定的。由此可见，研究网络的度分布对人们更好地分析目标

网络有着很重要的意义。关于数据集 Citeseer、DBLP 和 Cora 的度分布可视化，本文使用 MATLAB 计算出每个节点的度分布以及出现的次数。具体度分布可视化结果如图 3 所示。

如图 3 所示，横坐标表示度的大小，纵坐标表示该度值节点出现的次数。从图中可以发现，Cora 数据集中节点的最大度值小于 170, 但是每个度值出现的频率明显高于 DBLP 和 Citeseer 数据集，最高的度值出现了 570 余次。Citesee 和 DBLP 数据集中，度值小于 50 的节点具有高频率出现的现象，而度值在 50 至 200 之间的节点具有低频率出现的现象。因此，三个数据集中，大多数的节点的度值比较小只有很少一部分节点具有高度值。由此可知，Citeseer 数据集和 Cora 数据集不是高稠密的网络。

## 3.4 调参与分析

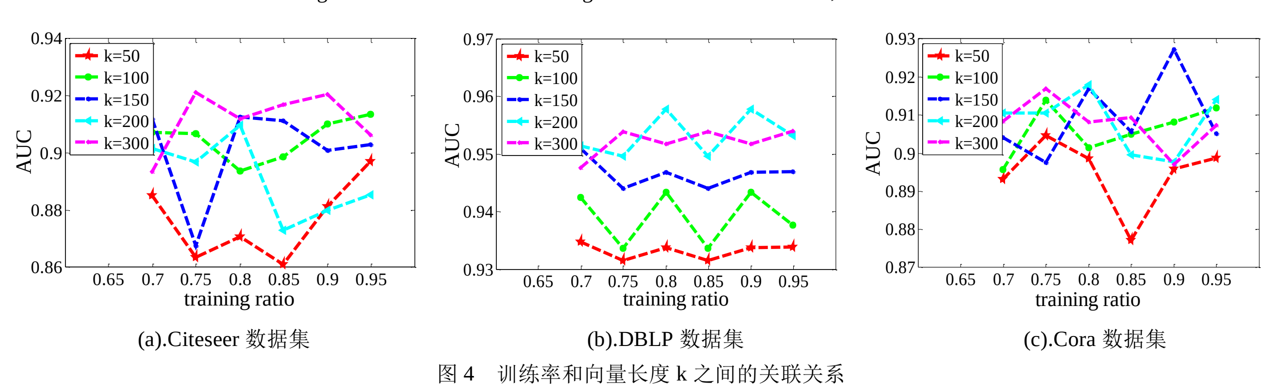
本文的实验中需要设置两个参数，分别是向量长度 $k$ 和训练集的训练比例 training ratio。设置训练集比例主要是为了分割出一部分测试数据，方便计算 AUC。在训练阶段，仅仅将分割出的训练集部分转换为邻接矩阵，然后再求得即将分解的目标矩阵。为了展示向量长度和训练率对 AUC 的影响，本文做了参数影响实验，具体的结果如图 4 所示。




(a).Citeseer 数据集 (b).DBLP 数据集 (c).Cora 数据集

图 3 在 citeseer, dblp 和 cora 数据集上的度分布可视化

Fig. 3 The visualizations of degree distribution on Citeseer, DBLP and Cora




(a).Citeseer 数据集 (b).DBLP 数据集 (c).Cora 数据集

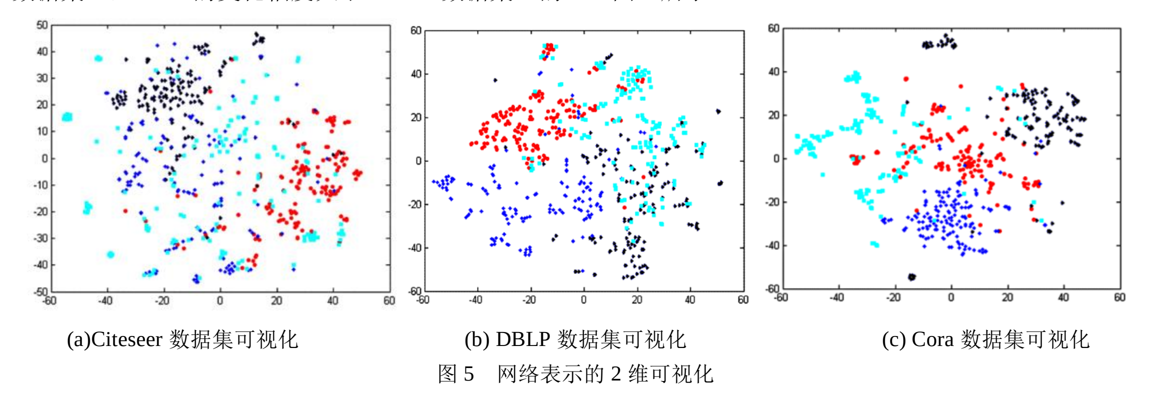
图 4 训练率和向量长度 k 之间的关联关系

Fig. 4 The correlations between training ratio and representation size

如图 4 所示，本文设置了向量长度为 50、100、150、200、300，设置了训练集的训练比例为 0.7、0.75、0.8、0.85、0.9、0.95。当向量长度为 50 时，Citeseer、DBLP 和 Cora 数据集上获得的 AUC 效果最差。当向量长度为 300 时，都获得了总体上比较好的性能。因为 Citeseer 和 Cora 网络是一个稀疏网络，所以当训练集比例为 0.9 时，AUC 获得了较好的性能。而 DBLP 是一个稠密的网络，因此，对于任何向量长度设置，随着训练集变化，AUC 的变化幅度非常小。但是，在 Citeseer 和 Cora 数据集上，AUC 的变化幅度大于 DBLP 数据集上的 AUC 变化。综上，可以总结出，对于稀疏网络，向量长度和训练集训练比例对 AUC 的影响较大，但是对于稠密网络，影响较小。

## 3.5 网络表示可视化

从 Citesser、DBLP 和 Cora 三个数据集中随机选取 4 个类别，每个类别随机选取 150 个节点。然后使用 T-SNE 可视化算法将每个数据集中的 600 个节点投影到 2 维的平面上，不同的类别用不用的颜色表示。投影后的 2 维可视化结果如图 5 所示。



(a)Citeseer 数据集可视化 (b) DBLP 数据集可视化 (c) Cora 数据集可视化

图 5 网络表示的 2 维可视化

Fig. 5 2D visualizations of network representations

从图 5 可以发现，网络表示的 2 维可视化结果展示出了较好的区域边界。因此，基于矩阵分解的网络表示学习算法训练得到的节点向量具有明显的可区分能力。可视化结果中，具有相同类标签的节点使用同一种颜色表示，然后使用降维算法将分类结果投影到 2 维平面上。在 Citeseer、DBLP 和 Cora 三个数据集的可视化结果中，可以发现，具有相同颜色的节点具有较为明显的聚类现象，投影在 2 维平面上，相似节点之间具有较近的距离。较为深层次地可以认为，使用基于矩阵分解的网络表示学习算法可以很好的学习和训练到网络的结构信息，使得具有相似网络结构的节点在表示空间中具有更近的距离；相反，具有相差较大的网络结构的节点在空间表示中具有更远的距离。可视化结果证实了本文所提出的 LPMF 算法训练得到的节点的表示具有聚类的功能。因此，在链路预测中，基于训练得到的网络表示基于聚类性质可更好的进行预测，而在网络表征中隐含的聚类属性也能辅助提升链路预测的精度。

### 3.6 案例研究

DBLP 是一个引文网络数据集，本文将该数据集中的论文分割为 4 个领域，数据库领域(来自于 SIGMOD、SIGMOD REC、ICDE、VLDB, EDBT、PODS、ICDT, DASFAA、SSDBM、CIKM、VLDB 等)、数据挖掘领域(来自于 KDD、ICDM、SDM、PKDD、PAKDD 等)、人工智能领域(来自于 IJCAI、AAAI、NIPS、ICML、ECML、ACML、IJCNN、UAI、ECAI、COLT、ACL、KR 等)、计算视觉领域 (来自于 CVPR、ICCV、ECCV、ACCV、MM、ICPR、ICIP、ICME 等)。在 DBLP 数据集中，本文通过随机函数随机选取一个目标节点，然后设置该目标节点的文本标题为“**Querying Object-Oriented Databases**”。随后通过计算余弦相似度值，得到与该标题节点相似度值最高的 5 个邻居节点，然后获取这 5 个节点的标题。在本节实验中，设置网络节点表示的长度为 100，训练率为 0.9。返回 5 个最相关节点标题的实验结果如表 3 所示。

表 3 案例研究

Table 3 Case study

| 论文标题 | 相似度 | 类别标签 |
|---|---|---|
| 1. A Powerful and Simple **Database Language** | 0.7476 | 数据库 |
| 2. A General Framework for The Optimization of **Object-Oriented Queries** | 0.7381 | 数据库 |
| 3. Towards an Effective Calculus for **Object Query Languages** | 0.7253 | 数据库 |
| 4. A Functional Execution Model for **Object Query Languages** | 0.7165 | 数据库 |
| 5. A **query Language** for Multidimensional Arrays Design Implementation and Optimization Techniques | 0.7065 | 数据库 |

如表 3 所示，通过网络表示的相似度计算，返回了与目标标题最相关的 5 个标题。另外，这 5 个最相关的标题与目标标题有很高的结构相似性，且都属于数据库领域。目标节点的标题是“**Querying Object-Oriented Databases**”，经过阅读该论文发现，该论文提出了一种新奇的结构化语言，可查询面向对象数据库。因此引用该论文或者被该论文引用的论文至少满足“查询语言”或者“数据库”中的一个条件。从表 3 中可以发现，本文算法所返回的 5 个最相关的标题包含有“**Query Languages**”或者“**Database**”。因此，基于矩阵分解的 DeepWalk 算法可有效的挖掘网络中的结构关联性，使得学习得到的网络表示通过相似的网络结构拥有更相近的空间距离。

## 4 结束语

本文首先通过证明 DeepWalk 网络表示学习算法，发现 DeepWalk 算法的实质即为矩阵分解。基于这个实事，本文提出了一种基于矩阵分解的 DeepWalk 路预测算法 LPMF。利用所提出的 LPMF 算法在 Citeseer、DBLP 和 Cora 三个真实的引文网络中进行无监督学习，实验结果表明 LPMF 算法在真实网络环境中的链路预测性能优异，其性能优于现有的大多数链路预测算法。另外，网络表示可视化实验得知了基于矩阵分解的 DeepWalk 算法训练得到的节点表示向量同样具有明显的聚类现象。案例研究实验证明了训练得到的节点的向量表示能够充分地反映网络的特征，使得具有相似结构的网络节点具有更相近的空间距离。综上，本文提出的 LPMF 链路预测算法是一种有效可行的算法，能够在实际网络的链路预测中发挥出较为出色的性能。在未来研究中，一方面，可将本文的算法与云计算等分布式框架相结合，满足超大规模的链路预测需求。另一方面，在分解本文的目标矩阵 $M$ 时，可使用融合外部信息的矩阵分解算法，从而更加充分的挖掘已知网络的相关特征，也可以将目标网络的其他重要信息融入到网络特征中。其相关的矩阵分解算法有 Inductive Matrix Completion、Dependent Probabilistic Matrix Factorization 等。

## 参考文献：

[1] Getoor L, Diehl C P. Link mining: a survey [J]. ACM SIGKDD Explorations Newsletter, 2005, 7(2): 3-12.

[2] Albert R, Barabasi A L. Statisrical mechanics of complex networks [J]. Reviews of Modern Physics, 2002, 74(51): 47-97.

[3] Dorogovtsev S N, Mendes J F F. Evolution of networks [J]. Advances in Physics, 2002, 51(4): 1079-1187.

[4] Leicht E A, Holme P, Newman M E. Vertex similarity in networks [J]. Physical Review E：Statistical Nonlinear & Soft Matter Physics, 2006, 73 (2): 026120.

[5] Zhang Luming, Gao Yue, Hon Chaoqun. Feature correlation hypergraph: exploiting high-order potentials for multimodal recognition [J]. IEEE Trans on Cybernetics, 2017, 44(8): 1408-1419.

[6] Yu Haiyuan, Braun P, Yildirim M A, *et al.* High-quality binary protein interaction map of the yeast interactome network [J]. Science, 2008, 322 (5898): 104-110.

[7] Stumpf M P, Thorne T, De S E, *et al.* Estimating the size of the human interactome [J]. Proceedings of the National Academy of Sciences of the United States of America, 2008, 105(19): 6959-6964.

[8] Xie Xiaoqin, Li Yijia, Zhang Zhiqiang, *et al.* A joint link prediction method for social network [C]//Proc fational Conference of Young Computer Scientists, Engineers and Educators. Berlin: Springer, 2015: 56-64.

[9] Schafer L, Graham J W, Missing data: our view of the state of the art [J]. Psychol. Methods, 2007, 7 (2): 147-152.

[10] Kossinets G. Effects of missing data in social networks [J]. Social Networks, 2003, 28 (3): 247-268.

[11] Kumar R, Novak J, Tomkins A. Structure and evolution of online social networks [M]//Link Mining: Models, Algorithms, and Applications. New York: Springer, 2006: 337-357.

[12] Gallagher B, Tong Hanghang, Eliassi-Rad T, *et al.* Using ghost edges for classification in sparsely labeled networks [C]//Proc of ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining, New York: ACM Press, 2008: 256-264.

[13] Dasgupta K, Singh R, Viswanathan B, *et al.* Social ties and their relevance to churn in mobile telecom networks [C]//Proc of the 11th International Conference on Extending Database Technology.2008: 668-677.

[14] Zadeh P M, Kobti Z. A knowledge based framework for link prediction in social networks [C]//Proc of International Symposium on Foundations of Information and Knowledge Systems. Austria: Springer, 2016: 255-268.

[15] Zhang Xue, Zhao Chengli, Wang Xiaojie, *et al.* Identifying missing and spurious interactions in directed networks [M]// Wireless Algorithms, Systems, and Applications. Harbin: Springer, 2014: 470-481.

[16] Guimerà R, Salespardo M. Missing and spurious interactions and the reconstruction of complex networks [J]. Proceedings of the National Academy of Sciences of the United States of America, 2009, 106 (52): 22073-8.

[17] Mering C V, Krause R, Snel B, *et al.* Comparative assessment of large-scale data sets of protein interactions [J]. Nature, 2002, 417 (6887): 399-403.

[18] Zhou Tao, Liu Linyuan, Zhang Yicheng. Predicting missing links via local information [J]. European Physical Journal B, 2009, 71 (4): 623-630.

[19] Leskovec J, Huttenlocher D, Kleinberg J. Predicting positive and negative links in online social networks [C]// Proc of International Conference on World Wide Web.New York: ACM Press, 2010: 641-650.

[20] Barabási A, Albert R. Emergence of scaling in random networks [J]. Science, 1999, 286 (5439): 509-512.

[21] Garlaschelli D, Capocci A, Caldarelli G. Self-organized network evolution coupled to extremal dynamics [J]. Nature Physics, 2008, 3 (11): 813-817.

[22] Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks [J]. Journal of the Association for Information Science & Technology, 2007, 58 (7): 1019–1031.

[23] Bianconi G. Entropy of network ensembles [J]. Physical Review E: Statistical Nonlinear & Soft Matter Physics, 2009, 79 (2): 036114.

[24] Liu Weiping, Lu Linyuan. Link prediction based on local random walk [J]. Europhysics Letters, 2010, 89(5): 58007-58012.

[25] Mikolov T, Chen Kai, Corrado G, *et al.* Efficient estimation of word representations in vector space [EB/OL]. [2017-09-05] https://arxiv. org/abs/1301. 3781.

[26] Mikolov T, Sutskever I, Chen Kai, *et al.* Distributed representations of words and phrases and their compositionality [J]. Advances in Neural Information Processing Systems, 2013, 26(14): 3111-3119.

[27] Berozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations [C]//Proc of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York: ACM Press, 2014: 701-710.

[28] Yang Cheng, Liu Zzhiyuan. Comprehend deepwalk as matrix factorization [EB/OL]. [2017-09-15] https://arxiv. org/abs/1301. 3781.

[29] Klein D J, Randic M. Resistance distance [J]. Journal of Mathematical Chemistry, 1993, 12(1): 81-95.

[30] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine [J]. computer network and ISDN system, 1998, 30(1): 107-117.

[31] Liu Weiping, Lü Linyuan. Link prediction based on local random walk [J]. Europhysics letters, 2010, 89(5): 58007-58012.

[32] François Lorrain, Harrison C. White. Structural equivalence of individuals in social networks [J]. Social Networks, 1977, 1(1): 67-98.

[33] Casey R G C B. Friends and neighbors [J]. Foreign Affairs, 2005, 46(3): 548-561.

[34] Zhou Tao, Lü Lingyuan, Zhang Yicheng. Predicting missing links via local information [J]. European Physical Journal B, 2009, 71(4): 623-630.

[35] Salton G, McGill M J. Introduction to modern information retrieval [J]. Program, 2004, 55 (3): 239-240.

[36] Jaccard P. Etude de la distribution florale dans une portion des Alpes et du Jura [J]. Bulletin De La Societe Vaudoise Des Sciences Naturelles, 1901, 37(142): 547-579.

[37] Sorensen T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons [J]. Biologiske Skrifter, 1957, 5(4): 1-34.

[38] Ravasz E, Somera A L, Mongru D A, *et al.* Hierarchical organization of modularity in metabolic networks [J]. Science, 2002, 297(5586): 1551-1555.

[39] Leicht E A, Holme P, Newman M E. Vertex similarity in networks [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2006, 73(2): 116-120.

[40] Katz L. A new status index derived from sociometric analysis [J]. Psychometrika, 1953, 18(1): 39-43.

[41] Fouss F, Pirotte A, Renders J M, *et al.* Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation [J]. IEEE Trans on Knowledge & Data Engineering, 2007, 19(3): 355-369.

[42] Li Ling. Link prediction based on random walks [J]. Journal of Computational Information Systems, 2015, 11(5): 1757-1764.

[43] Chebotarev P, Shamis E. The matrix-forest theorem and measuring relations in small social groups [J]. Automation & Remote Control, 2006, 58(9): 1505-1514.

[44] Sun Duo, Zhou Tao, Liu Jianguo, *et al.* Information filtering based on transferring similarity [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2009, 80(2): 017101.

[45] Fortunato S, Flammini A, Menczer F. Scale-free network growth by ranking [J]. Physical Review Letters, 2006, 96(21): 218701.

[46] Lü Linyuan. Link prediction in complex networks: a local naive Bayes model [J]. EurophysicsLetters, 2011, 96(4): 48007.

[47] Clauset A, &Amp C M, Newman M E J. Hierarchical structure and the prediction of missing links in networks [J]. Nature, 2008, 453(7191): 98-101.

[48] Redner S. Networks: teasing out the missing links [J]. Nature, 2008, 453 (7191): 47-48.

[49] Yang Cheng, Liu Zhiyuan, Zhao Deli, *et al.* Network representation learning with rich text information [C]//Proc of International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2015: 2111-2117.

[50] Tu Ccunchao, Zhang Weicheng, Liu Zhiyuan, *et al.* Max-margin deepwalk: discriminative learning of network representation [C]// Proc of International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2016: 3889-3895.

[51] Yang Cheng, Sun Maosun, Liu Zhiyuan, *et al.* Fast network embedding enhancement via high order proximity approximation [C]//Proc of

International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2017: 3894-3900.

[52] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization [J]. Advances in Neural Information Processing Systems, 2014, 3 (17): 2177-2185.

[53] Yu H F, Jain P, Kar P, *et al*. Large-scale multi-label Learning with Missing Labels [C]//Proc of International Conference on Machine Learning. New York: ACM Press, 2013: 593-601.